International Journal of Combined Research & Development (IJCRD)
eISSN:2321-225X; pISSN:2321-2241 Volume: 2; Issue: 6; June -2014

SCALED HYBRID PERMUTATION NETWORK FOR MULTIPROCESSOR SYSTEM

[1]SACHINKUMAR H PUJAR, [2]RAMALINGAM H M

[1,2]ELECTRONICS AND COMMUNICATION ENGINEERING, MITE, India
Email: [1]sachinkumarpujar4@gmail.com, [2]ramalingam@mite.ac.in

Abstract: The project presents the design of on chip permutation network for the multiprocessors with the data lossless, predictable latency, guaranteed bandwidth and in order delivery. The proposed network contains the pipelined circuit switching and dynamic path setup scheme. The circuit switching provides the guarantee of this permuted data and enables the stacking of multiple networks with saving power and area. Round robin arbiter is used along with hexagonal mapping. A 0.13µ silicon design validates feasibility and efficiency for the network.

Keywords: Guaranteed throughput, multistage interconnection network, pipelined circuit switching, traffic permutation, network on-chip.

I. INTRODUCTION

Now a days drift of multiprocessor system-on-chip (MPSoC) design being interconnected with on-chip network is currently emerging for various applications of parallel processing, scientific computing and so on[6].In the permutation traffic where a traffic pattern in which each input sends traffic to exactly one output and each output receives traffic from exactly one input, is one of the important traffic occur in general-purpose MPSoC, for example, polynomial, sorting, and fast Fourier Transform(FFT) computations cause shuffled permutation ,whereas matrix transposes or corner-turn operations exhibit transpose permutation[6].and fast Fourier transform (FFT) computations cause shuffled permutation , whereas matrix transposes or corner-turn operations exhibit trans-pose permutation [6]. Recently application –specific MPSoCs targeting flexible Turbo/LPDC decoding have been developed and they exhibit arbitary and concurrent traffic permutations due to multimode and multi standard feature. In fact many MPSoC applications in real time require the advantage of having data lossless, guaranteed bandwidth, predictable latency , in order delivery is critical for such permutation traffics. Different on-chip networks use different routing algorithms such as dimension-ordered routing and minimal adaptive routing. The application aware routings also used in these networks which are configured before running the applications and can be implemented as source routing or distributed routing .But such application-aware routings cannot efficiently handle the dynamic changes of a permutation pattern, which is exhibited in many of the application phase[6].But permutation networks suffer from the area and power criteria. Reviewing on-chip networks (supporting either full or partialpermutation) with regard to their implementation shows that most the networks employ a packet switching mechanism to deal with the encounter the conflict of permuted data[6].

For general purpose networks regular topologies would be used like mesh, torus which are intuitively feasible for physical layout in a2-D chip. But for high wiring irregularity and the large router radix of indirect topologies such as benes or butterfly[5] pose a challenge for physical implementation. Because arbitary permutation pattern with its intensive load on individual source –destination pairs stresses the regular topologies and that may lead to throughput degradation.

The proposed network design consists of an on-chip permutation network to support guaranteed throughput of permuted traffics under arbitary permutation. Unlike conventional packet switching-switching approaches our on-chip permutation network to support guaranteed throughput of permuted traffics under arbitary permutation. The proposed design employs pipelined circuit switching with dynamic path–setup scheme under a multistage network topology. The dynamic path-setup tackles the challenge of runtime path arrangement for conflict-free permuted data. Preconfigured path data paths enable a throughput guarantee. By removing the excessive over-head of queuing buffers, a compact implementation is achieved and stacking multiple networks to support concurrent permutations in run-time is feasible.

The rest of this paper is organised as follows. Section II presents the on-chip network design with its dynamic path-setup scheme to support runtime path arrangement. Section III gives the implementation details and reports a proof-of-concept test chip.and finally, Section IV concludes this paper and outlines the further researches.

II. PROPOSED ON_CHIP NETWORK DESIGN

As discussed in section I, the key idea of proposed on-chip network design is based on a pipelined circuit-switching approach with a dynamic path-setup scheme supporting runtime path arrangement. Before mentioning the dynamic path-setup scheme, the network topology is first discussed. Then design of switching nodes are presented.

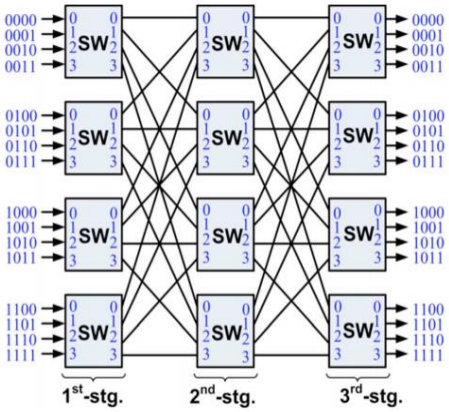

Fig. 1. Proposed On-Chip network topology with port addressing scheme.

A. On-Chip Network Topology

Clos network, a family of multistage networks, is applied to build scalable commercial multiprocessors with thousands of nodes in macro systems[7].A typical three-stage Clos network is defined as C(n, m, p) ,where n represents the number of inputs in each of p first-stage switches and p is number of second-stage switches. In order to support a parallelism degree of 16 as in most practical MPSoCs [3],we proposed to use C(4,4,4) as a topology for the designed network(Fig 1). This network has a rearrangeable property[11] that will provide all permutations between its inputs and outputs. The choice of the three-stage Clos network with modest number of middle-stage switches is to minimize implementation cost, whereas it still enables a rearrangeable property for the network.

A pipelined circuit-switching scheme is designed for use with the proposed network. The scheme has three phase setup, transfer and release. A dynamic path-setup scheme supporting the runtime path arrangement occurs in the setup phase. In order to support this circuit-switching scheme, a switch-by-switch interconnection with its handshake signals is prosed, as shown in (Fig.2)

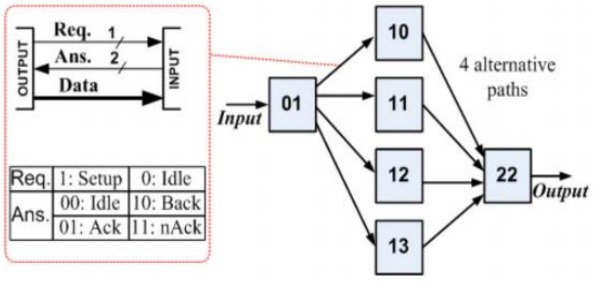

Fig.2. Switch by switch interconnection and path diversity capacity

The bit format of the handshake includes a 1-bit Request (Req) and a 2-bit Answer (Ans).Req=1 is used when a switch requests an idle link leading to the corresponding downstream switch in the setup phase. The Req=1 is also kept during data transfer along the set up path. A Req=0 denotes that the switch releases the occupied link. This code is also used in both setup and release phases. An Ans=01(Ack) means that the destination is ready to receive the data from the source. When the Ans=01 propagates back to the source, it denotes that the path is setup, then a data transfer can be started immediately. An Ans=11(n Ack) is reserved for end-to-end flow control when the receiving circuit is not ready to receive data due to being busy with other tasks, or overflow at the receiving buffer etc. An Ans =10 Back means that the link is blocked.

B. Dynamic Path Setup to Support Path Arrangement

A Dynamic path-setup scheme is the key point of the proposed design to support a runtime path arrangement when the permutation is changed. Each path setup, which starts from an input to find a path leading to its corresponding output, is based on a dynamic probing mechanism.[2],[9]. In which a probe (or setup flit)is dynamically sent under a routing algorithm in order to establish a path towards the destination. Exhausted profitable backtracking is implemented to use route the probe in the network work. A path arrangement with full permutation consists of sixteen path setups, whereas path arrangement with partial permutation may consists of a subset of sixteen path setups.

As designed in this network, each input sends a probe containing a 4bit output address to find an available path leading to the target output. During the search, the probe moves forwards when it finds a free link and moves backwards when it faces a blocked link. By means of no-repetitive movement, the probe finds path between its input and corresponding idle output. The EPB-based path

setups scheme is designed with a set of probe routing algorithms. The following example describes how the path setup works to find an

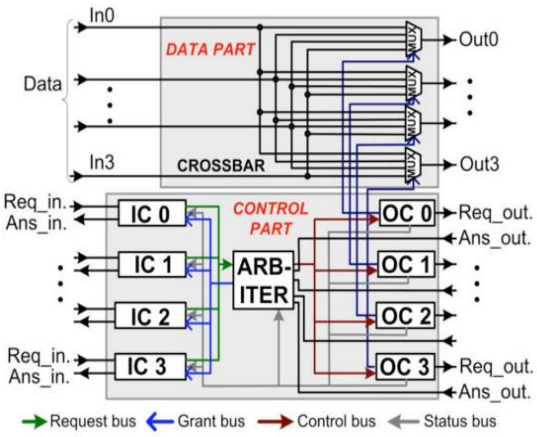

Fig.3. Common switch architecture

available path by using the set of path diversity shown in Fig 2.It is assumed that a probe from a source (e.g. an input of switch 01) is trying to setup a path to target destination (e.g. an available output of switch 22).First, the probe will non-repetitively try paths through the second stage switches in the order of 10-11-12-13.Assuming that link01-10 is available, the probe first try this link (Req=1).then arrives at switch 10.

i. If link 10-22 is available, the probe arrives at switch 22 and meets the target output. An Ans=Ack then propagates back to the input to trigger the transfer phase.

ii. If link 10-22 is blocked, the probe will move back to 01(Ans=Back) and link 01-10 is released Req=0.From switch 01, the probe can try the rest of idle links leading to the second-stage switches in the same manner. By means of moving back when facing blocked links and trying others ,the probe can dynamically set up the path in runtime back when facing blocked links and trying others, the probe can dynamically set up the path in runtime in a conflict avoidance manner.

C. Switching Node Designs

Three kinds of switches are been designed for the proposed on chip network. These switches are all based on a common switch architecture shown in Fig 3. With the only difference being in the probe routing algorithms. This common architecture has basic components INPUT CONTROLS, OUTPUT CONTROLS, ARBITER and a CROSSBAR. Incoming ed through the data paths to save on wiring costs.

The Arbiter has two functions:, the cross connecting the Ans-Outs and the ICs through the grant bus, and second , as a referee for the requests from the ICs. When an incoming probe arrives at an input, the corresponding ICs observes the output status bus, and requests the ARBITER to grant it access to the corresponding OC through the Request bus. When accepting this request, the ARBITER cross-connects the corresponding As-Out with the IC through the Grant bus with its first function. With the second function, the ARBITER, based on a pre-defined priority rule, resolves contention when several ICs request the same free output. After this resolution only one IC is accepted, whereas the rest are answered as facing a blocked link (i.e similar to receiving an (Ans=Back).The IC is implemented with finite state machine. The probe routing algorithm and the operation of the switches are controlled according to this FSM implementation in the ICs[9].In order to support the probing path setups, ICS are implemented with different probe routing algorithms depending on its switch stage. The probe contains the 4-bit address of the switch stage. The probe contains the 4 bit address of the destination i.e $D_3D_2D_1D_0$(see fig 1 for the addressing scheme).The three routing algorithms for the switches in the first ,the second and the third stages. In the first stage, the switch tries the free outputs in a non-reparative.

This implementation avoids repetitively avoids repetitively searching the same path that may result in lovelocks. The second and third stage switches rely on the two most significant bits significant bits D_3D_2 and the two least significant bits D_1D_0 of the destination address, respectively, to route the probe. Depending on the availability of the desired output or the feedback from the downstream switch, the IC in a given switch will change its FSM state and reply to the upstream switches accordingly.

The OCs work as re-timing stages for the commands from ARBITER placed on the control bus and control the CROSSBAR. The CROSSBAR is a 4×4 full connecting matrix designed with output multiplexers.

The ICs and the ARBITER are clocked with the rising and the falling edges of the clock, respectively. By this implementation probing is dynamically processed by the switching one clock cycle basis.

As denoted in fig 3 ,the control part of switches performs the dynamic EPB-based path setup, whereas the data part simply provides configured paths for guaranteed circuit-switched data. This meets the target of designing the circuit – switched switches to support EPB-based path set up in C (4,4,4) in network.

To validate if the designed network works as desired works as desired, test bench is applied to test the capability of realizing full permutation with

six-teen path setups To avoid a path setup interfering others during the search and incurring a rearrangement of existing paths, a delay is set between the path setups launched one-by-one in a sequences in the test bench. This is to ensure the previous path setups launched one-by-one in a sequence in the test bench. This is to ensure that the previous path setup is completed before a new one is launched one-by-one in a sequence in the test bench. This is to ensure that the previous path setup is completed before a new one is launched. As calculated on path diversity graph shown in Fig 2, the worst-case path setup needs 14steps(hops) of moving its probe back and forth to search a path. Each step of moving the probe needs two cycles, as derived from cycle-accumulator rate design model. Hence, we set the delay to a delay to a value of 28 cycles (i.e 14*2=28) Arranging a full permutation requires 448 cycles to complete. By this setting, we simulate and validate the success of the design in arranging over ten different sets of 1000 random full permutations.

III. IMPLEMENTATION AND RESULTS

Whenever inputs have to be applied ,clock should be asserted by the clock (do not forcing the value) and for the first cycle reset should be forced to “1”and next cycles we can apply “0” and so on. Whenever the Req bus to “1” is applied, It will grant it for next cycle, then the acknowledgement will be sent. Then for the next cycle it will observe the status of Request bus and responds to next grant bus by asserting grant to “1”.When all the Requests are raised to 1,it looks for the Request which is been raised to “1”.Like this the arbiter is going to provide the grant ,based on their priority.

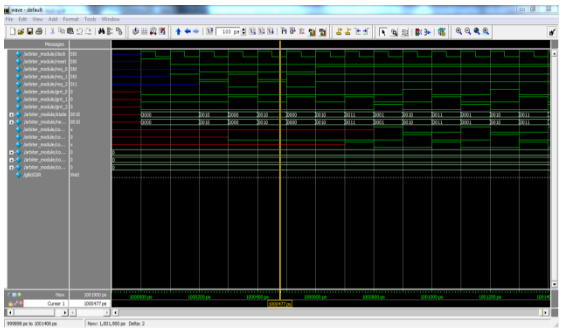

Fig.4 Simulated waveform of arbiter

TABLE 1

Comparison with other related On-Chip Networks

Design	1	2	3	4
Number of input× outputs	16×16	16×8	16×8	16×16
Topology	De Bruijn	Butterfly	Benes 2N	3 stage Clos
Datawidth(bit)	20	32	32	16(16*2)
Min/Max Latency Cycl	NA	5/21	8/22	16/28
Evaluation Level	Post synth	Post synthes	Post synthes	Post Si
Measured BW	-	-	-	28-35

CONCLUSION

The proposed network design will help in accessing with multiple processors. With using parallel circuit ,which mainly conserves the amount of power and space the will be consumed with stacking multiple network. So the dynamic path setup mainly helps in upgrading the traffic network giving the user guaranteed permutation. So the paper helps in making the scientific computations, transpose permutations easier.

REFERENCES

[1] S. Borkar, “Thousand core chips—A technology perspective,” in Proc.ACM/IEEE Design Autom. Conf. (DAC), 2007, pp. 746–749.
[2] P.-H. Pham, P. Mau, and C. Kim, “A 64-PE folded-torus intra-chipcommunication fabric for guaranteed throughput in network-on-chip based applications,” in Proc. IEEE Custom Integr. Circuits Conf.(CICC), 2009, pp. 645–648.
[3] C. Neeb, M. J. Thul, and N. Wehn, “Network-on-chip-centric approach to interleaving in high throughput channel decoders,” in Proc. IEEE Int.Symp. Circuits Syst. (ISCAS) , 2005, pp. 1766–1769.
[4] H. Moussa, A. Baghdadi, and M. Jezequel, “Binary de Bruijn on-chip network for a flexible multiprocessor LDPC decoder,” in Proc. ACM/IEEE Design Autom. Conf. (DAC) , 2008, pp. 429–434.
[5] H. Moussa, O. Muller, A. Baghdadi, and M. Jezequel, “Butterfly and Benes-based on-chip communication networks for multiprocessor turbo decoding,” in Proc. Design, Autom. Test in Euro. (DATE), 2007, pp. 654–659.
[6] S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, “An 80-tile sub-100-w TeraFLOPS processor in 65-nm CMOS,” IEEE J. Solid-State Circuits, vol. 43, no.
1, pp. 29–41, Jan. 2008.

[7] W. J. Dally and B. Towles , Principles and Practices of Interconnection Networks:. San Francisco, CA: Morgan Kaufmann, 2004.
[8] N. Michael, M. Nikolov, A. Tang, G. E. Suh, and C. Batten, "Analysisof application-aware on-chip routing under traffic uncertainty," in Proc.IEEE/ACM Int. Symp. Netw. Chip (NoCS) , 2011, pp. 9–16.